```
procedure INIT-PDA
{Invoked when the router comes up. }
begin
   Initialize all tables;
   call PDA;
end INIT-PDA
```

```
procedure PDA
{Executed at each router i. Invoked when an event occurs}
begin
   (1) call NTU;
   (2) call MTU; /* Updates T^i */
   (3) if (there are changes to T^i) then
             Compose an LSU message consisting of topology
             differences using add, delete
             and change link entries;
       end if
   (4) Within a finite amount time, send the
       LSU message to all neighbors;
end PDA
```

FIG. 1

procedure NTU
begin
(1) if (LSU message is received from a neighbor k) then
(1a) Update neighbor table T_k^i. That is, add links, delete links or change links according to the specification of each entry in the LSU;
(1b) Run Dijkstra's shortest path algorithm on the resulting topology T_k^i; /*This results in finding minimum distances from k to all other nodes in T_k^i. Note T_k^i is a tree*/
(1c) Update D_{jk}^i with new distances in T_k^i;
end if
(2) if (adjacent link (i, k) is up) then
Update l_k^i and send an LSU message to the neighbor k with link information of all links in its main topology table T^i;
endif
(3) if (cost of an adjacent link (i, k) changed)then
Update l_k^i;
endif
(4) if (adjacent link (i, k) failed)then
Update l_k^i and clear the table T_k^i;
endif
end NTU

FIG. 2

```
procedure MTU at router i
begin
        (1) oldT^i ← T^i:    /* Save copy */
        (2) if (node j occurs in at least one of T_k^i) then
                add j to the main topology table T^i
            end if
        (3) for each node j in T^i do
              MIN ← min {D_jk^i + l_k^i | k ∈ N^i};
              let p be such that MIN = (D_jp^i + l_p^i);
              /* Neighbor p is the preferred neighbor for
              destination j. Ties are broken in favor of
              lower address neighbor */
            done
        (4) for each j in T^i and its preferred neighbor p do
                Copy all links (j,n) from T_p^i to T^i;
                /* i.e., copy all links in T_p^i for which
                j is the head node */
             done
        (5) Update T^i with information of each l_k^i;
        (6) Run Dijkstra's shortest path algorithm on T^i
             and remove those links in T^i that are not
             part of the shortest path tree;
        (7) Update D_j^i with new distances in T^i;
        (8) Compare oldT^i with T^i and note all differences;
end MTU
```

FIG. 3

```
procedure MPDA at router i
{invoked when an event occurs}
begin
        (1) call NTU;
        (2) if (node is in PASSIVE state) then
               (2a) call MTU; /* update T^i and D_j^i */
               (2b) F D_j^i ← min{F D_j^i, D_j^i};
            endif
        (3) if (node is in ACTIVE state and the
                    last ACK is received) then
               (3a) temp_j^i ← D_j^i; set node to PASSIVE state;
               (3b) call MTU to update T^i;
               (3c) F D_j^i ← min{ temp_j^i, D_j^i}
            endif
        (4) S_j^i ← {k|D_j^i < F D_j^i};
        (5) if (changes occur in T^i)then
               Set node to ACTIVE state;
            endif
            if (no changes occur in T^i and the event is
                    the last ACK) then
               Set node to PASSIVE state;
            endif
        (6) if (there are changes to T^i) then
                Compose anew LSU with the topology
                changes expressed as add link,
                delete link and change link;
            end if
        (7) if (input event received is an LSU message)then
               Add the ACK entry to newly composed LSU;
            endif
        (8) Send the new LSU message.
end MPDA
```

FIG. 4

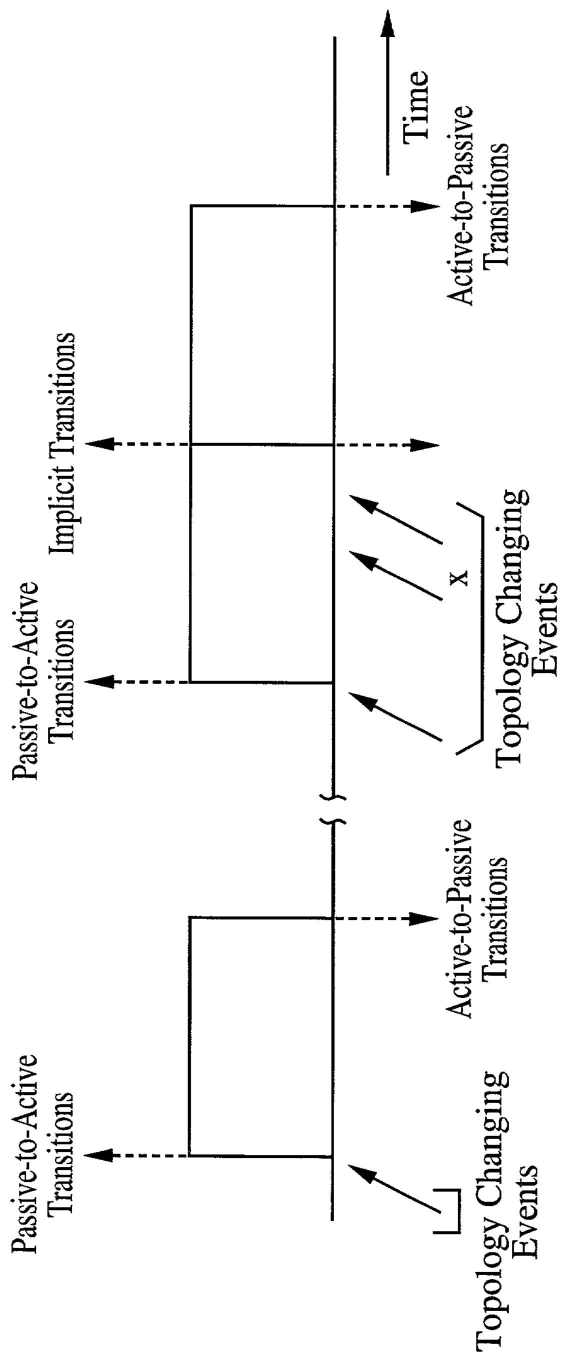

Time
Active-to-Passive Transitions
Implicit Transitions
Passive-to-Active Transitions
x
Topology Changing Events
Active-to-Passive Transitions
Passive-to-Active Transitions
Topology Changing Events

FIG. 5

Procedure IH

begin

(1) $\forall k \notin S_j^i, \phi_{jk}^i \leftarrow 0;$

(2) if $(|S_j^i| = 1)$ then

$\forall k \in S_j^i, \phi_{jk}^i \leftarrow 1;$

endif

(3) if $(|S_j^i| > 1)$ then

$$\phi_{jk}^i \leftarrow \frac{1 - \dfrac{D_{jk}^i + l_k^i}{\sum_{m \in S_j^i} (D_j^i + l_m^i)}}{(|S_j^i| - 1)}, \ \forall k \in S_j^i;$$

endif

end IH

FIG. 6

Procedure AH

begin

(1) $D^{ij}_{min} \leftarrow min\ \{D^i_{jk} + l^i_k | k \in S_j\}$;

(2) let $D^{ij}_{min} = (D^i_{jk_0} + l^i_{k_0})$

// that is, k_0 be the neighbor

that offers the minimum)

(3) foreach $k \in S^i_j$ do

$a^i_{jk} \leftarrow D^i_{jk} + l^i_k - D^{ij}_{min}$;

done

(4) $\Delta \leftarrow \frac{1}{2} min\ \{\frac{\phi^i_{jk}}{a^i_{jk}} | k \in S^i_j \wedge a^i_{jk} \neq 0\}$;

(5) foreach $k \neq k_0 \wedge k \in S^i_j$ do

$\phi^i_{jk} \leftarrow \phi^i_{jk} - \Delta \times a^i_{jk}$;

done

(6) foreach $k = k_0$ do

$\phi^i_{jk} \leftarrow \phi^i_{jk} + \sum_{q \in S_{jk}} \Delta \times a^i_{jq}$;

done

end AH

FIG. 7

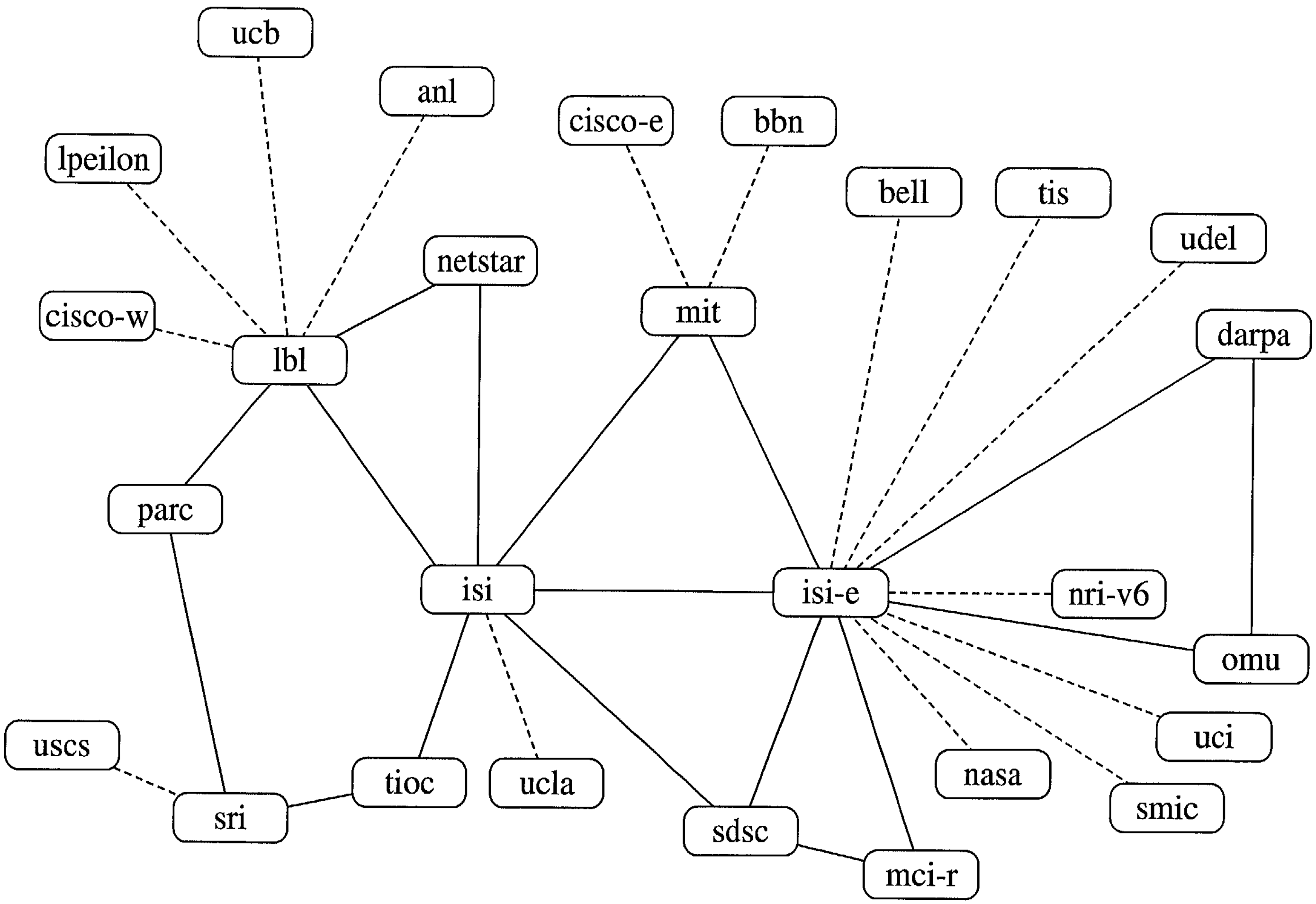

ucb
anl
lpeilon
cisco-e
bbn
bell
tis
udel
netstar
cisco-w
lbl
mit
darpa
parc
isi
isi-e
nri-v6
omu
uci
uscs
tioc
ucla
nasa
smic
sri
sdsc
mci-r

FIG. 8

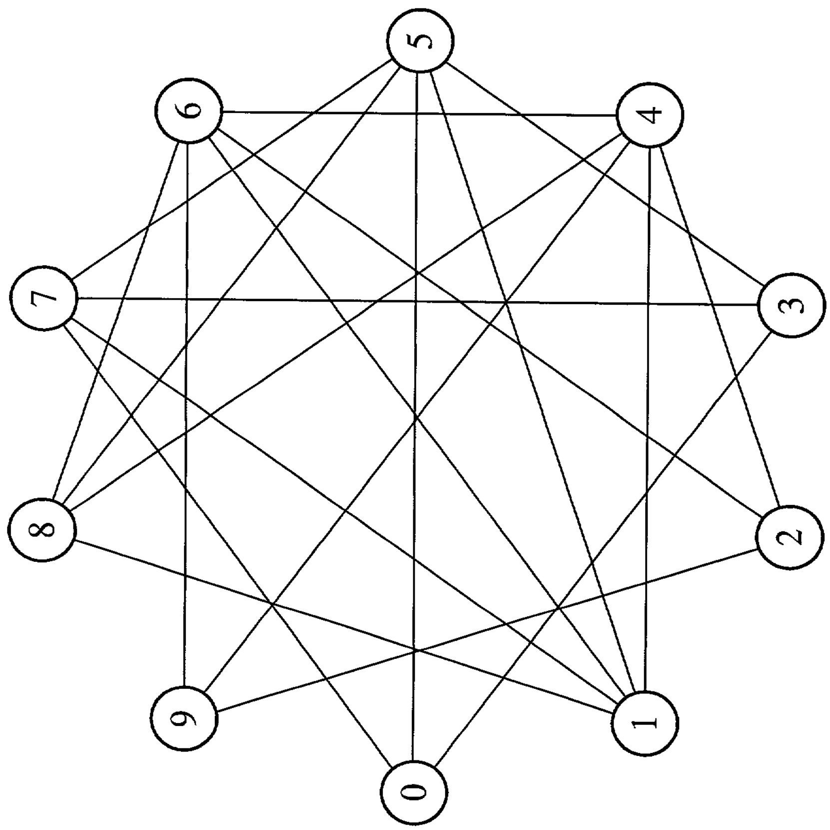
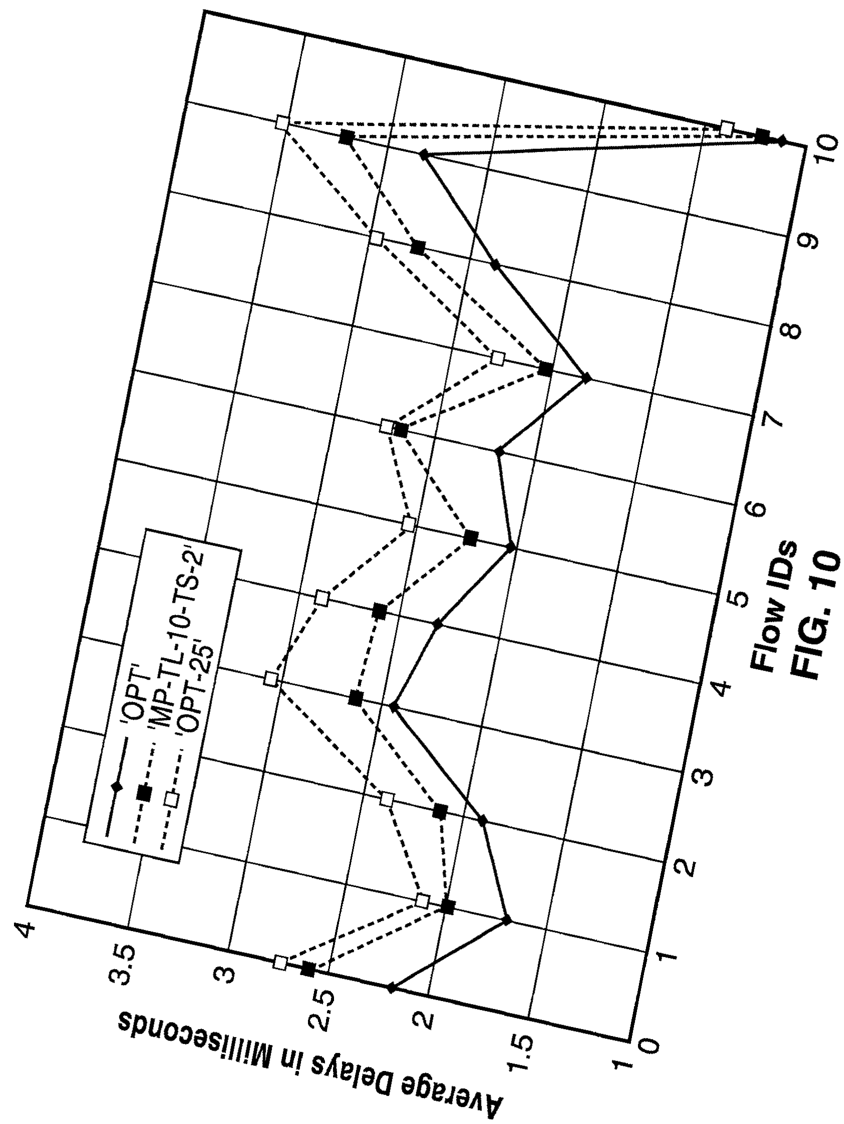


FIG. 9



'OPT'
'MP-TL-10-TS-2'
'OPT-25'
Average Delays in Milliseconds
1
1.5
2
2.5
3
3.5
4
0
1
2
3
4
5
6
7
8
9
10
Flow IDs

FIG. 10

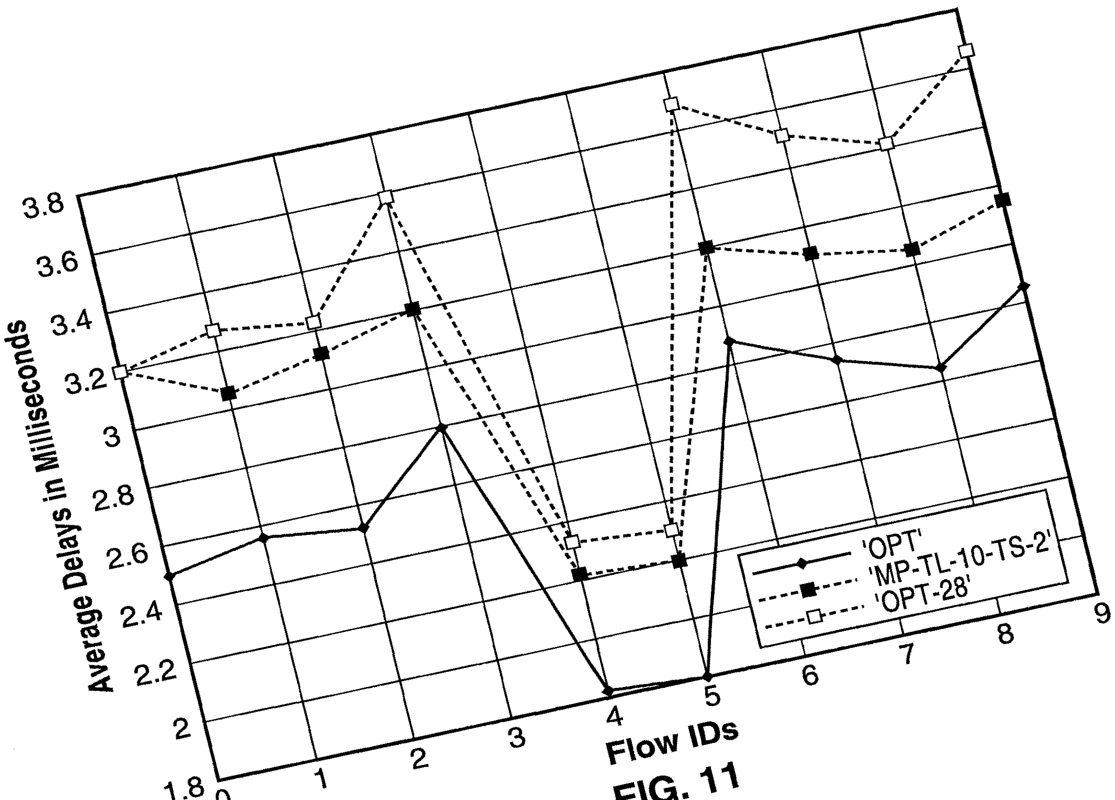

Average Delays in Milliseconds
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
1.8
0
1
2
3
4
5
6
7
8
9
Flow IDs
'OPT'
'MP-TL-10-TS-2'
'OPT-28'

FIG. 11

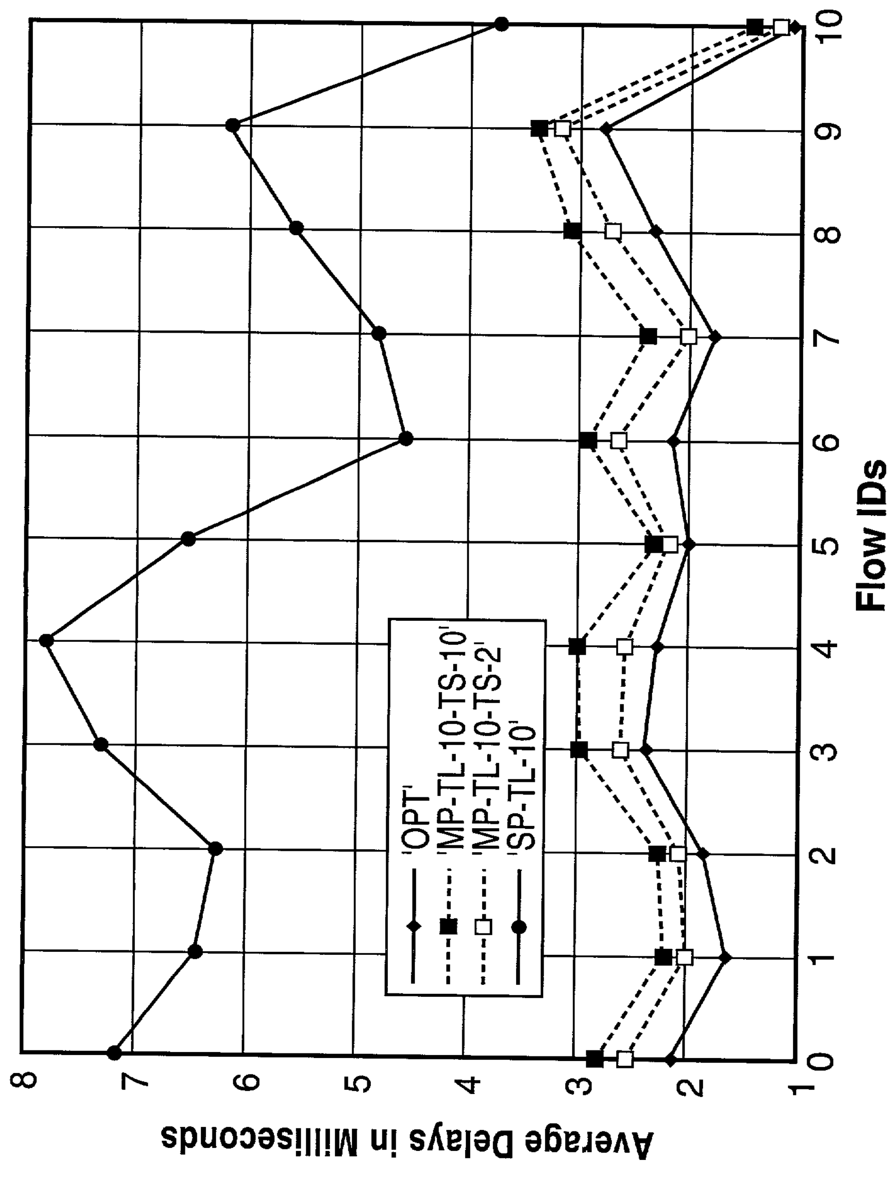
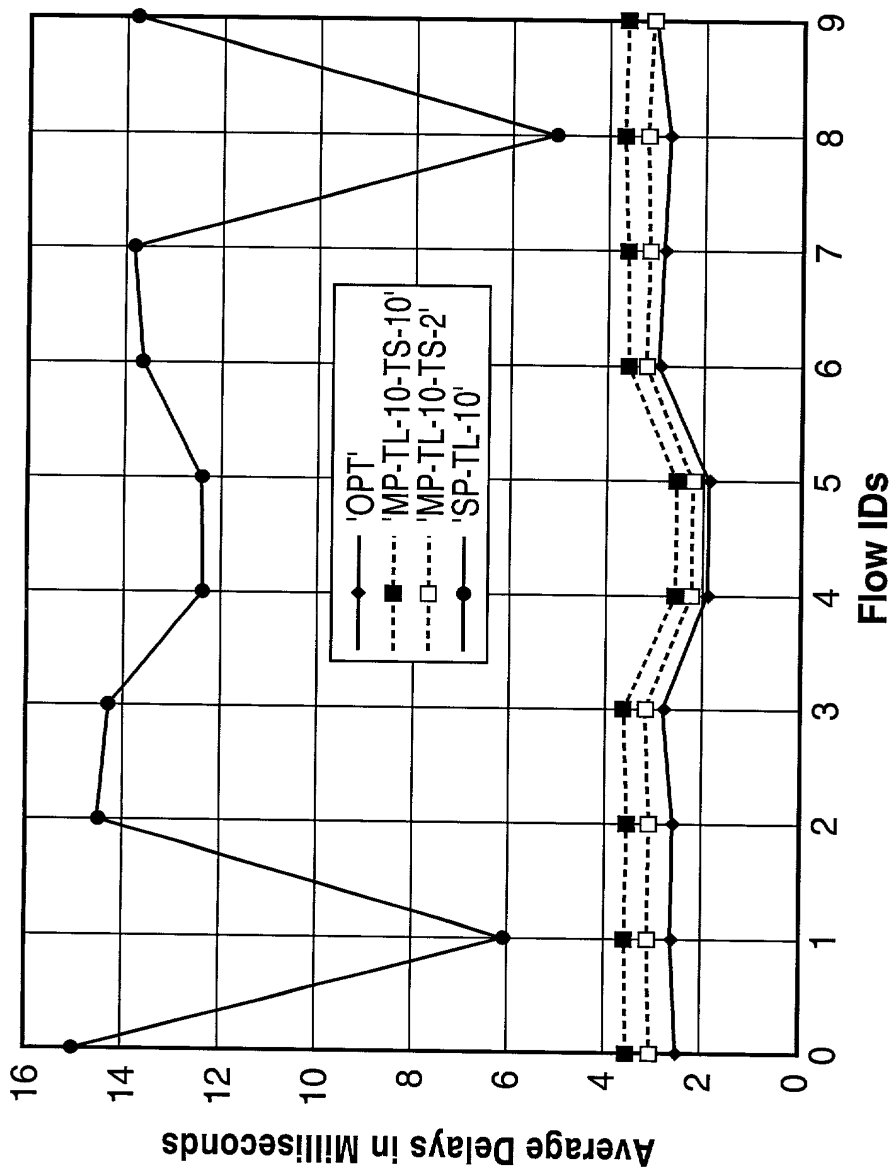

Average Delays in Milliseconds
1
2
3
4
5
6
7
8
0
1
2
3
4
5
6
7
8
9
10
Flow IDs
'OPT'
'MP-TL-10-TS-10'
'MP-TL-10-TS-2'
'SP-TL-10'

FIG. 12



Average Delays in Milliseconds
0
2
4
6
8
10
12
14
16
Flow IDs
0
1
2
3
4
5
6
7
8
9
'OPT'
'MP-TL-10-TS-10'
'MP-TL-10-TS-2'
'SP-TL-10'

FIG. 13

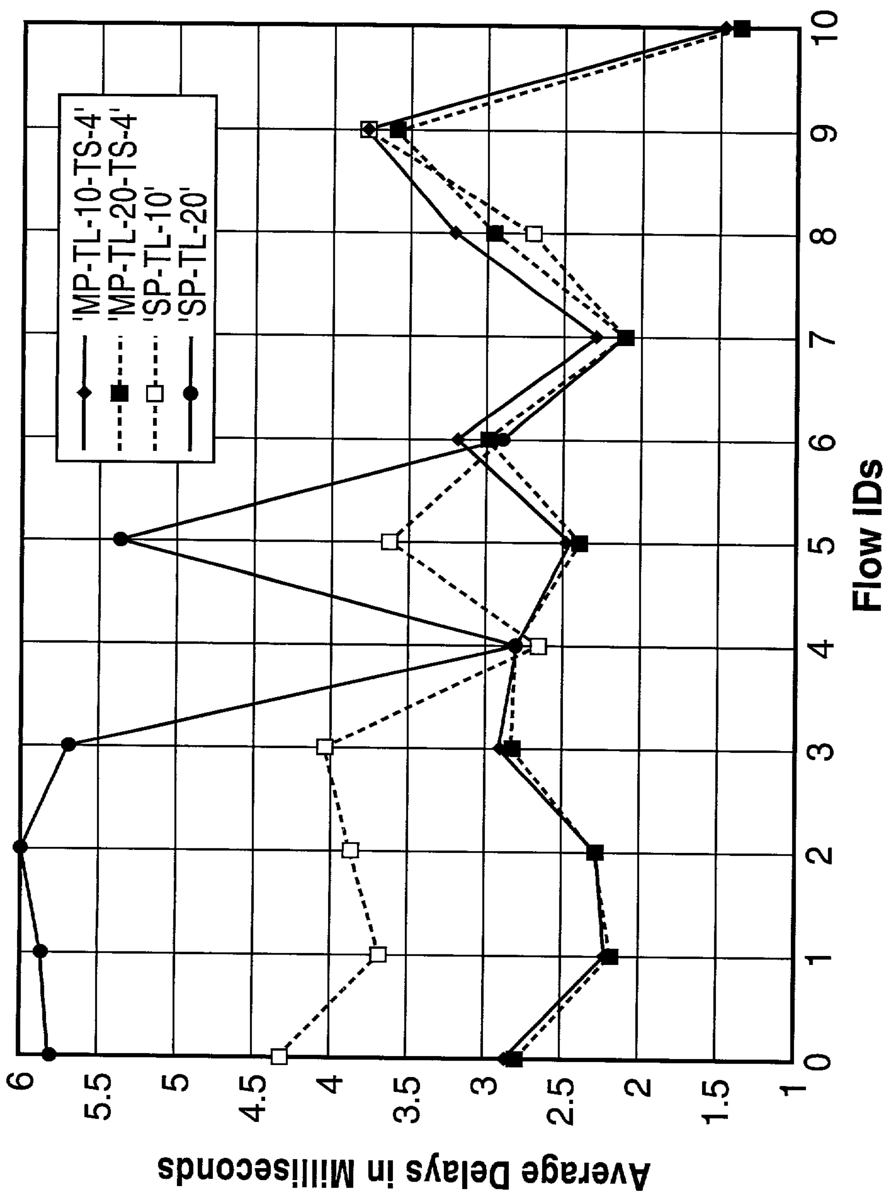

'MP-TL-10-TS-4'
'MP-TL-20-TS-4'
'SP-TL-10'
'SP-TL-20'
Average Delays in Milliseconds
6
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0
1
2
3
4
5
6
7
8
9
10
Flow IDs

FIG. 14

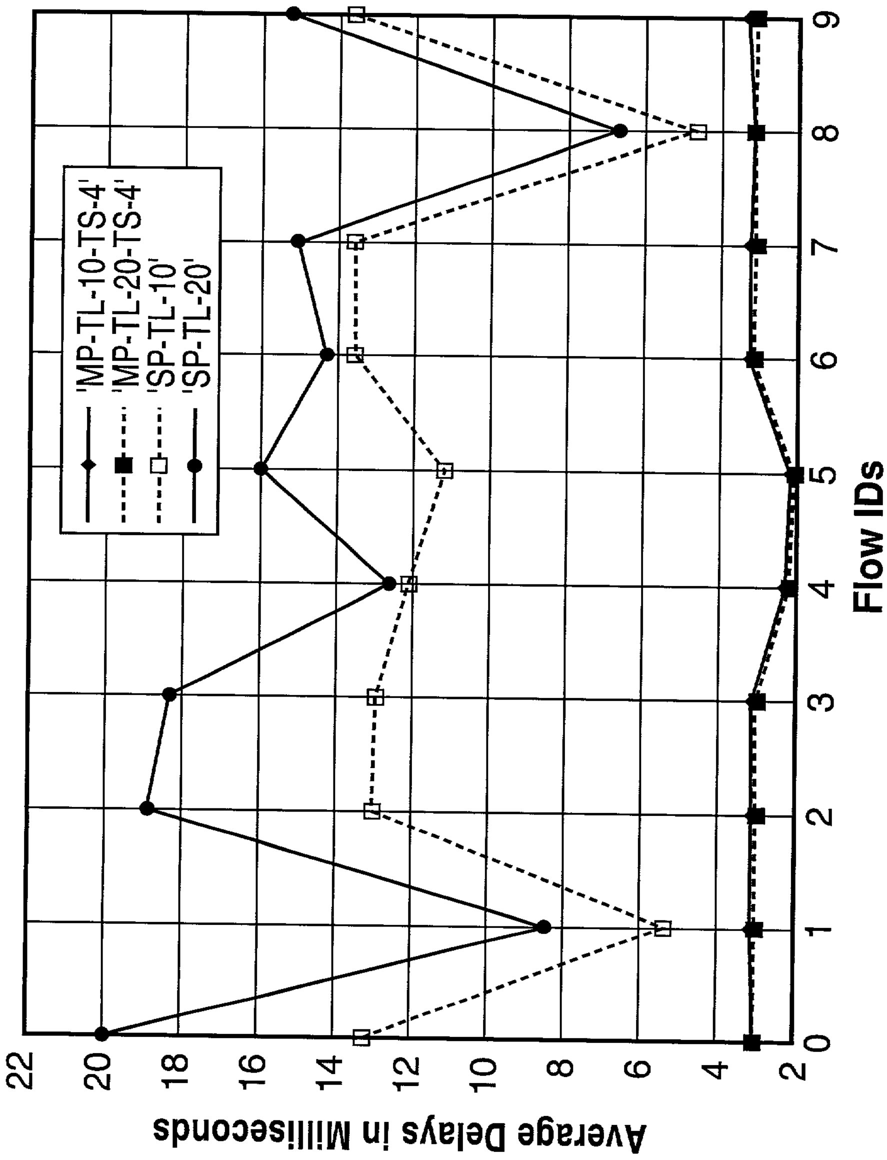

'MP-TL-10-TS-4'
'MP-TL-20-TS-4'
'SP-TL-10'
'SP-TL-20'
Average Delays in Milliseconds
22
20
18
16
14
12
10
8
6
4
2
0
1
2
3
4
5
6
7
8
9
Flow IDs

FIG. 15

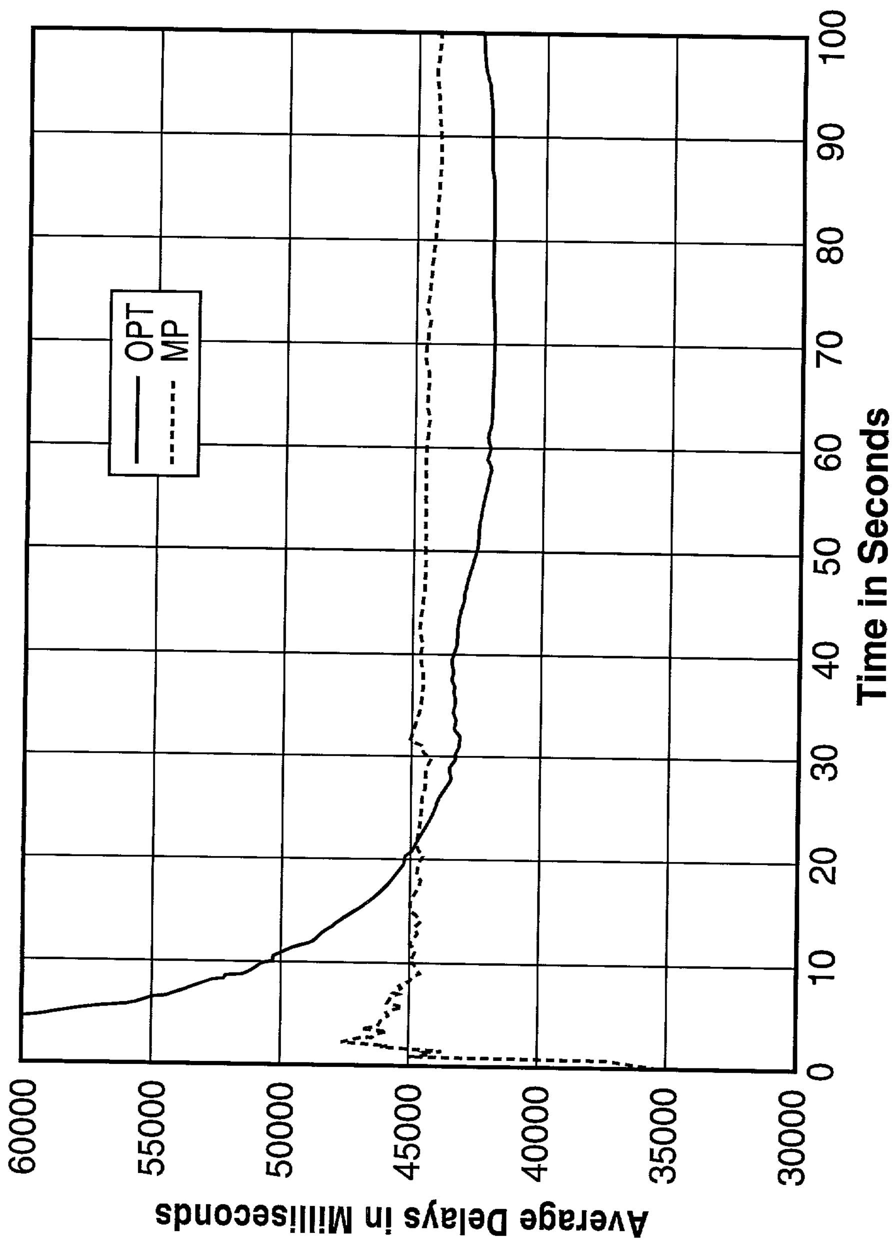

Average Delays in Milliseconds
60000
55000
50000
45000
40000
35000
30000
Time in Seconds
0
10
20
30
40
50
60
70
80
90
100
OPT
MP

FIG. 16

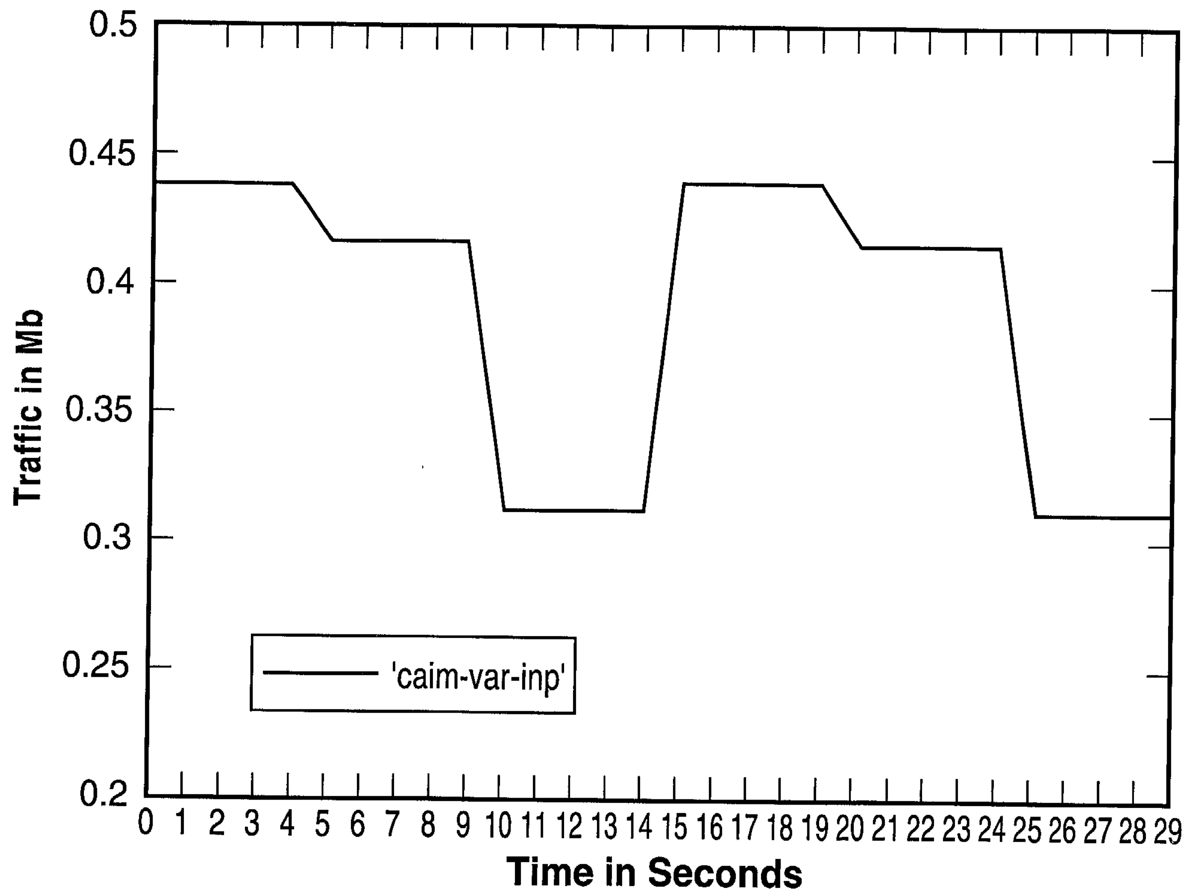

0.5
0.45
0.4
0.35
0.3
0.25
0.2
Traffic in Mb
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
Time in Seconds
'caim-var-inp'

FIG. 17

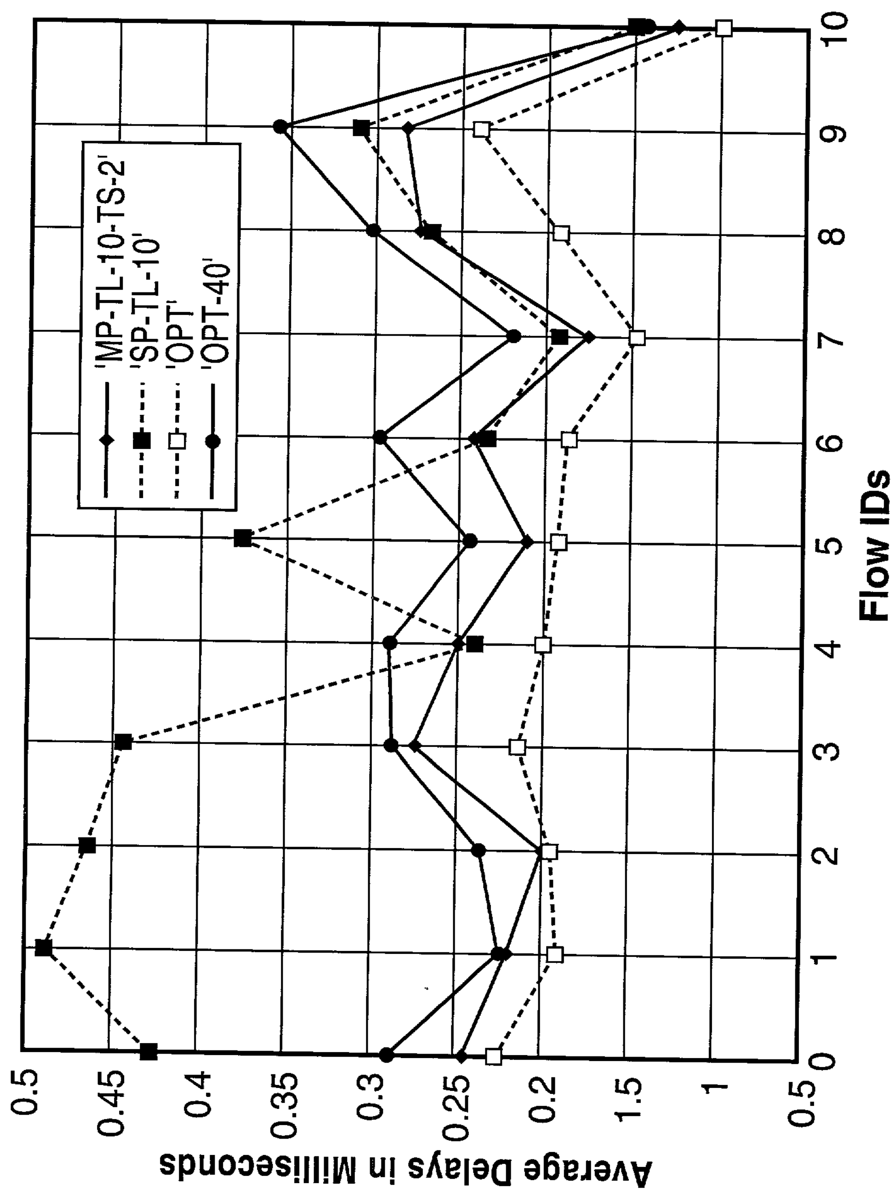

'MP-TL-10-TS-2'
'SP-TL-10'
'OPT'
'OPT-40'
Average Delays in Milliseconds
0.5
0.45
0.4
0.35
0.3
0.25
0.2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10
Flow IDs

FIG. 18

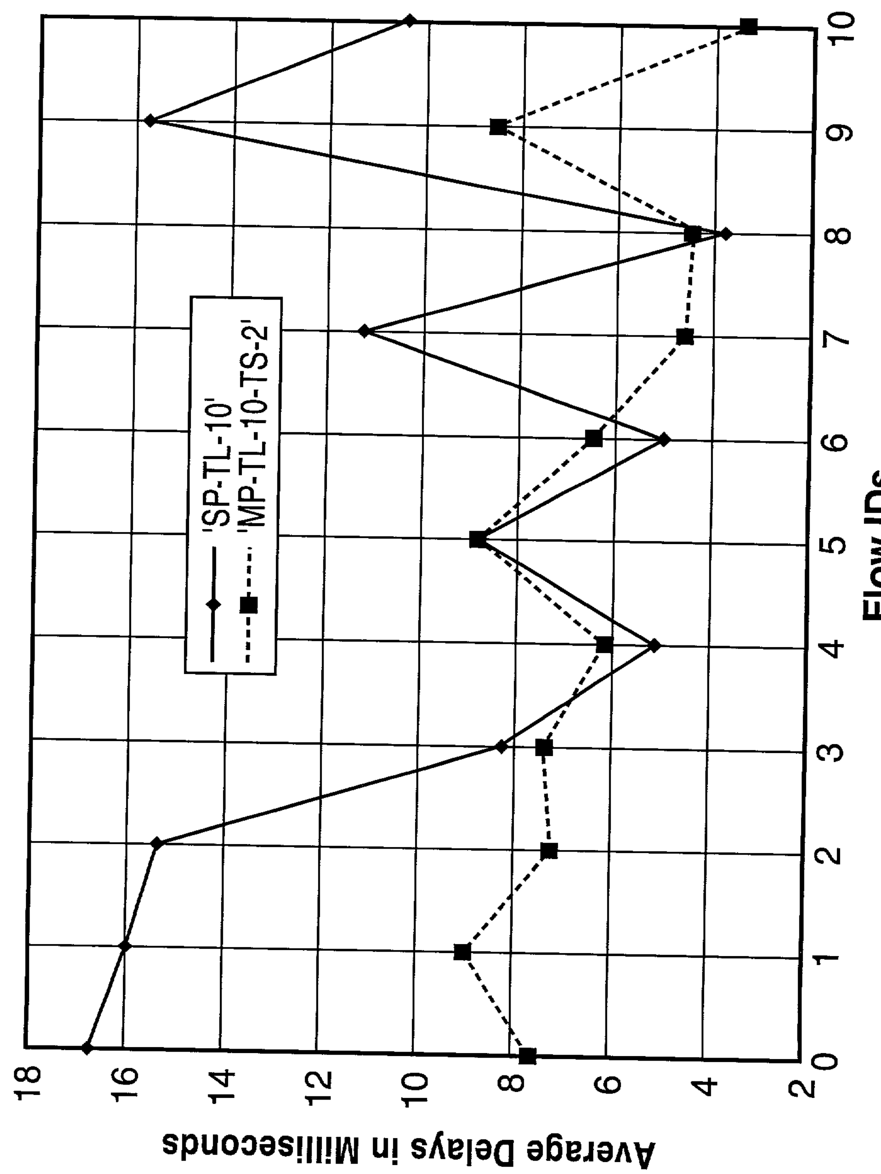

'SP-TL-10'
'MP-TL-10-TS-2'
Average Delays in Milliseconds
Flow IDs
0
1
2
3
4
5
6
7
8
9
10
2
4
6
8
10
12
14
16
18

FIG. 19